Proceedings of Machine Learning Research 1–19

Scale-Equivariant UNet for Histopathology Image Segmentation

Yilong Yang Yilong.Yang@soton.ac.uk
Srinandan Dasmahapatra sd@ecs.soton.ac.uk
Sasan Mahmoodi sm3@ecs.soton.ac.uk
University of Southampton, University Road, Southampton, SO17 1BJ, United Kingdom

Abstract

Digital histopathology slides are scanned and viewed under different magnifications and stored as images at different resolutions. Convolutional Neural Networks (CNNs) trained on such images at a given scale fail to generalise to those at different scales. This inability is often addressed by augmenting training data with re-scaled images, allowing a model with sufficient capacity to learn the requisite patterns. Alternatively, designing CNN filters to be scale-equivariant frees up model capacity to learn discriminative features. In this paper, we propose the Scale-Equivariant UNet (SEUNet) for image segmentation by building on scale-space theory. The SEUNet contains groups of filters that are linear combinations of Gaussian basis filters, whose scale parameters are trainable but constrained to span disjoint scales through the layers of the network. Extensive experiments on a nuclei segmentation dataset and a tissue type segmentation dataset demonstrate that our method outperforms other approaches, with much fewer trainable parameters.

Keywords: UNet, Scale, Equivariant, Segmentation

1. Introduction

Pathologists diagnosing biopsy samples view histopathology slices at different magnifications by controlling the microscope's objective revolver. Neural network based decision support for digital pathology take as input digital images scanned from glass slides. Specimen slides scanned at different medical institutions may use different objective magnifications to digitalize specimen slides, resulting in whole slide images (WSI) being at different scales. For example, images provided by the CRAG dataset Awan et al. (2017) are in 20× magnification; For the DigestPath-2019 dataset Li et al. (2019), images are in 40× magnification. Models such as Convolutional neural networks (CNNs) trained on images at a specific scale generally can not generalise to other scales, which greatly restricts the applicability of computer-aided diagnosis models.

CNNs have dominated the computer vision field since the proposal of the AlexNet (Krizhevsky et al., 2012). The most widely adopted strategy to cope with scale variation in unseen data is introducing scale augmentation during training CNNs, where training samples are randomly scaled before being fed into the network. Other attempts such as scale selection (Girshick et al., 2014) and scale fusion (Kokkinos, 2015) also help to circumvent scale changes. However, these methods lack explicit mechanisms to model scale information. Some works such as Kanazawa et al. (2014); Marcos et al. (2018); Xu et al. (2014) achieve scale equivariance by resizing the input or filter, but these methods are computationally expensive since they rely on tensor resizing and image interpolation. Other ways of

© Y. Yang, S. Dasmahapatra & S. Mahmoodi.

generating filters of different sizes include Bekkers (2019); Sosnovik et al. (2020); Zhu et al. (2022), parameterising filters by a trainable linear combination of a family of predefined, fixed multi-scale basis functions (Hermite, Fourier, B-Splines). Such methods, however, require that both the scale of basis functions and the size of filters should be fixed, once the network has been initialised. The work presented in Pintea et al. (2021) shows that hard-coding the scale hyper-parameters in the network can be restrictive, while learning the scale parameter is especially beneficial when dealing with inputs at multiple resolutions.

In this paper, we introduce the Scale-Equivariant UNet (SEUNet), which demonstrates superior generalisation performance on image datasets at different scales when compared with the conventional CNN model and other scale-equivariant models. The main characteristics of out work are as follows: 1) We parameterise convolutional filters with learnable Gaussian derivative filters, instead of using a set of pre-calculated, fixed filter basis. 2) We impose range constraints on learnable scale parameters to ensure coverage of multiple scales, while allowing them to be tuned within disjoint intervals. This frees up model capacity to find an optimal set of scale parameters that adapt to training samples by back-propagation.

2. Related Work

In recent years, group equivariance as an inductive bias for CNNs has influenced the design of several architectures including scale-equivariant convolutional networks. Worrall and Welling (2019) propose deep scale-space (DSS) based on the theory of scale-space and semi-groups to model transformation properties of images under scale transformations, modelling filter rescaling by dilation. However, the DSS is restricted only to integer scale factors, and therefore does not cover a continuous range of scale variations. To extend DSS to arbitrary scales, Sosnovik et al. (2021a,b) propose Discrete Scale Convolution (DISCO) wherein the equivariance error between the non-integer scale factor with its two nearest integer scale factors is minimised. In Scale-Equivariant Steerable Networks (SESN) (Sosnovik et al., 2020), filters are parameterised by trainable linear combination of pre-calculated Hermite basis functions. These are defined in the continuous scale domain and then projected on pixel grids for a set of given scale factors. Although SESN and DISCO allow the use of arbitrary scale factors, the best set of scale factors are dataset and network dependent and need to be carefully chosen to maximise model performance.

Gaussian scale-space theory (Lindeberg, 1994) represents an image as a one-parameter family of gradually smoothed signals, in which the fine scale details are successively suppressed by convolving the image with a set of re-scaled Gaussian filters and Gaussian derivative filters. Lindeberg (2022) proposes a Gaussian derivative network in which every convolutional filter is constructed as a linear combination of Gaussian derivative filters. The architecture presented in Lindeberg (2022) is only evaluated on image classification tasks, for which global scale invariance is key to predictive accuracy. For image segmentation tasks, the output map should scale in proportional to the input, making scale equivariance a necessary property. Similarly, Pintea et al. (2021) learn linear combinations of N-th order Gaussian derivative filters to create the N-Jet convolutional layer. Unlike Lindeberg (2022) and Sosnovik et al. (2020) where the scale parameters (σ) are fixed, the σ and sizes of the filters in the N-Jet layer are learned from the data; this frees the network architect from

searching and setting scale-related parameters for datasets and networks. However, the σ is shared by all filters in a layer, thus limiting the representational capacity of a N-Jet layer.

Our work extends Lindeberg (2022) from image classification to image segmentation with while also allowing the σ of each layer to be learnable similar to Pintea et al. (2021). Furthermore, we set the scale factors σ to lie in disjoint ranges through the layers of the network.

3. Methodology

Scale transformations and scale equivariance. The scaling operator S_s is defined on a function (image) f thus:

$$(S_s f)(x) = f(s^{-1}x),\ s > 0. \tag{1}$$

For Φ a family of feature mapping operators, scale equivariance means that the scaling transformation should commute with the feature mapping operation according to

$$\Phi'(S_s f) = S_s(\Phi(f)), \tag{2}$$

Where Φ' denotes some feature map operators within the same family Φ that operates on the image re-scaled by factor of s. We refer to cases with $s > 1$ as up-scalings and to cases with $s < 1$ as down-scalings.

3.1. Parameterising convolutional filters, layer by layer

The 1D Gaussian filter at scale σ is written as $G(x;\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$ which can be extended to 2D isotropic Gaussian filters as $G(x,y;\sigma) = G(x;\sigma)G(y;\sigma)$. Then the 2D Gaussian derivatives can be defined by the product of the partial derivatives on x and on y:

$$G^{i,j}(x,y;\sigma) = \frac{\partial^{i+j} G(x,y;\sigma)}{\partial x^i \partial y^j} = \frac{\partial^i G(x;\sigma)}{\partial x^i} \frac{\partial^j G(y;\sigma)}{\partial y^j} \tag{3}$$

Filter construction. In conventional CNNs, a bank of filters F^l of size $[C_l, C_{l-1}, h, w]$ is used to map an input image f^0 or feature map $f^{l-1} \in \mathbb{R}^{C_{l-1}\times H\times W}$ into $f^l \in \mathbb{R}^{C_l \times H \times W}$ by convolution. Here $l \geq 1$ is the layer index, (H, W) and (h, w) denotes the size of the feature map and the size of filter, respectively. Padding is applied therefore the size of the feature output remains the same as the input. We compute scale-space feature maps by convolving with groups of filters at different scales, with each convolutional filter a linear combination of Gaussian derivative filters:

$$F^l_k(c_l, x, y, \sigma^l_k; c_{l-1}) = \sum_{i,j\geq 0}^{i+j\leq N} \alpha^l_{i,j,c_l,c_{l-1}} G^{i,j}(x,y;\sigma^l_k), \tag{4}$$

where $\alpha^l_{i,j,c_l,c_{l-1}} \in \mathbb{R}$ are learnable, and independent of k which indexes scale settings within a layer. We describe these next, first for the first layer from image to features, and then how features are composed in subsequent layers. In detail, the C_l channels in F^l are divided into γ groups denoted by $F^l_k \in \mathbb{R}^{\frac{C_l}{\gamma}\times C_{l-1}\times h\times w}$, $k \in \{1,\ldots,\gamma\}$. The first layer maps the input

image f^0 into $f^1 = F^1 \star f^0$, $f^1 \in \mathbb{R}^{\gamma \times \frac{C_1}{\gamma} \times H \times W}$, by convolving with filters $F^1_k(c_1, x, y, \sigma^1_k; c_0)$ at position (x, y), input channel $c_0 \in \{\mathtt{r}, \mathtt{g}, \mathtt{b}\}$ and output channel $c_1 \in \{1, \cdots, \frac{C_1}{\gamma}\}$. The first dimension in f^1 represents the scale axis, with γ scales indexed by k.

Note that the subscript k of σ^l_k denotes that the scale parameter varies across groups in the same convolutional layer l, for all l, but is shared across filters in the same group. The $\alpha^l_{i,j,c_l,c_{l-1}}$ in equation (4) does not have a group index k in its subscript, as we share these learnable weights between groups, in order to ensure that the convolution kernels generated in different groups are consistent in shape, and do not mix separated scale factors, thus ensuring scale equivariance. When $\gamma = 1$, the F^l degenerates to N-Jet convolutional filter in which the σ is shared for the complete layer (Pintea et al., 2021). We visualise the constructed multi-scale filters in Appendix B.

Scale convolution in hidden layers. For layers $l \geq 2$ feature maps are divided into γ groups $f^{l-1} \in \mathbb{R}^{\gamma \times \frac{C_{l-1}}{\gamma} \times H \times W}$, each representing the response to a specific scale in $f^{l-1}_k \in \mathbb{R}^{\frac{C_{l-1}}{\gamma} \times H \times W}$. We again use equation (4) to construct γ groups of filters $F^l = [F^l_1, \cdots, F^l_\gamma]$, $F^l_k \in \mathbb{R}^{\frac{C_l}{\gamma} \times \frac{C_{l-1}}{\gamma} \times h \times w}$ to convolve with f^l_k. After the first layer, we define the network architecture to have $f^l_k = F^l_k \star f^{l-1}_k$: in subsequent layers each group of filters acts only on a subset of scale-matched channels. This is in contrast to the first layer ($l = 1$) where the filters act on the entire image to generate $f^1_k := (f^1)_k = F^1_k \star f^0$, with all colour channels contributing to the γ scale-specific channels in f^1 indexed by k. We thus have the learnable coefficients α and σ^l_k range over channel indices

$$\{\alpha^l_{i,j,c_l,c_{l-1}} | c_0 \in \{\mathtt{r}, \mathtt{g}, \mathtt{b}\}, c_l = 1, \ldots, \frac{C_l}{\gamma}\} \text{ and } \{\sigma^l_k | k = 1, \ldots, \gamma\}. \tag{5}$$

Thus, the propagation of information captured by composition of layer-wise convolutions does not mix information from different scales. The restriction on network connections to scale-matched layer outputs is designed to maintain equivariance to input rescaling at layer outputs under composition. Although acting F^l_k on the entire f^l can be another option, an attempt in Sosnovik et al. (2020) shows that introducing inter-scale interaction also introduces extra equivariance error, and leads to lower performance. We further tune the successive scale factors σ^l_k to track the increase in the receptive field with depth.

3.2. Imposing range constraints on σ^l_k

The trainable parameters in equation (5) in the filters include the scale parameters σ^l_k learned during back-propagation. However, leaving them to be tuned completely freely may lead to a problem: all σ^l_ks in the same layer may have the same value, which means constructed filters are redundant, limiting the scale diversity of filters. As our original intention is that the network can achieve scale equivariance by learning multi-scale convolutional filters, we introduce the following constraints to separate σ^l_k values to lie in disjoint intervals.

$$\sigma^l_k(x) = \frac{a^l_k - b^l_k}{2} \tanh x + \frac{a^l_k + b^l_k}{2}, \; a^l_k > b^l_k, \; b \geq 0 \tag{6}$$

where a^l_k and b^l_k are hyper-parameters for the upper and lower bounds for σ of filters at the lth layer and the kth group. x is a trainable real variable. By setting an appropriate set of

a_k^l and b_k^l: multi-scale filters can be constructed as per equations (4). Once σ_k^l is known, the following formula used in Pintea et al. (2021) is employed to determine the size τ_k^l of filters f_k^l:

$$\tau_k^l = 2\left\lceil 2\sigma_k^l \right\rceil + 1,\ \sigma_k^l > 0 \tag{7}$$

This enables us to train the size of the receptive field. In the encoder path of the UNet (layer 1-8), we gradually increase σ, to increase receptive field size. This is in keeping with Lindeberg (2022). In the decoder path (layer 11-18), we gradually decrease σ. Layer 9-10 are the bottleneck layers. An ablation study with regard to the setting of σ_k^l can be found in Appendix 5 which demonstrates the benefit of imposing range constraints on σ_k^l.

3.3. Parallelising training by simultaneous optimisation of multiple loss functions

As described in section 3.1, filters in each layer are divided into γ groups and each group of filters operate only on an non-overlapping subset of feature maps with no inter-scale feature interactions. Thus, features learned by these γ groups of filters can be trained in a mutually independent fashion. A penultimate convolution layer for feature fusion creates a score that is passed to a softmax function, followed by the calculation of loss function. This, however, mixes multi-scale information and destroys the scale equivariance of the features. Therefore, to train all groups of filters simultaneously while maintaining equivariance between multi-scale features, we propose to minimise a weighted combination of multiple loss functions, with each of them acting only on a single group of filters. In detail, given the ground truth y and feature map f_k^{L-1} that is produced by filters F_k^{L-1} in a L-layer network, a 1×1 convolution with softmax activation is used to map f_k^{L-1} into a probability map $\hat{y}_k$ for each of C classes for every pixel in the $H \times W$ image. The loss function used to train F_k^{L-1} is the norm cross-entropy loss:

$$l_k(y, \hat{y}_k) = -y\log(\hat{y}_k),\ y \in \{0,1\}^{C\times H\times W}, \hat{y}_k \in \mathbb{R}^{C\times H\times W},\ k = 1, \ldots, \gamma. \tag{8}$$

The overall loss function is defined as:

$$L = \sum_{k=1}^{\gamma} \widetilde{\eta}_k l_k(y, \hat{y}_k),\ \widetilde{\eta}_k = \frac{\eta_k + \frac{1}{\gamma}}{\sum_{k=1}^{\gamma}(\eta_k + \frac{1}{\gamma})},\ \text{ where } 0 \le \eta_k \le 1,\ \sum_{k=1}^{\gamma}\eta_k = 1,\ \sum_{k=1}^{\gamma}\widetilde{\eta}_k = 1. \tag{9}$$

η_k is a weighting factor that assigned to F_k to characterise the relative importance between scales. It is quite plausible that the loss function is minimised by a dominating contribution from a specific scale indexed by k, driving all other η_k to zero, a phenomenon called competitive exclusion. It is to maintain some contribution from features acquired at multiple scales that we introduce the additive constant $(1/\gamma)$ in $\widetilde{\eta}_k$. This constrains the trainable $\widetilde{\eta}_k$ to be in the range $[\frac{1}{2\gamma}, \frac{\gamma+1}{2\gamma}]$. In practice, we initialise η_k to $\frac{1}{\gamma}$ and use the *softmax* function to normalise η_k to guarantee $\sum_{k=1}^{\gamma}\eta_k = 1$. Figure 1 shows the entire structure of the model proposed here.

3.4. Final Prediction Generation

The SEUNet generates probability maps $\hat{y}_{k,n}$, $k = 1, \ldots, \gamma$ from learned filters with different scales/sizes for each pixel n. For each pixel n, let $\hat{y}_{k,n,c}$ be the probability of predicting class

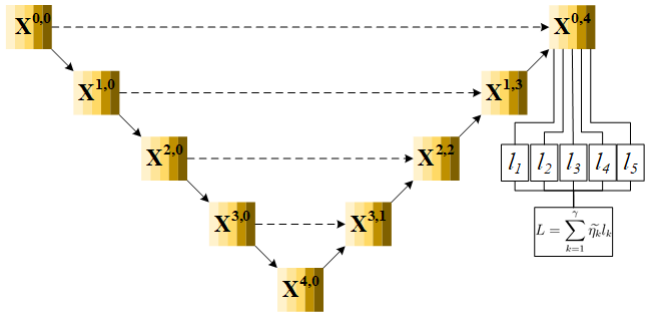


Figure 1: The architecture of the Scale-Equivariant UNet (SEUNet) model proposed here. Each square node in the graph represents a convolution block that consists of two convolutional layers. In each square node, five rectangles in different colours denotes five groups of filters that are parameterised by different σ_k^ls, but share the same α. All filters with the same colour form an independent sub-network, and all sub-networks have their own prediction and loss functions $\{l_k \mid k \in \{1, \cdots, 5\}\}$. All sub-networks can be trained simultaneously in an end-to-end fashion by minimising the combined loss function.

$c \in C$ by classifier indexed by scale k. Given an image with unknown scale information and these γ probability maps, we explore the following strategies to generate the final segmentation map.

Arithmetic mean ensemble. For each pixel n the final segmentation map is obtained from $\arg\max_c (1/\gamma) \sum_k \hat{y}_{k,n,c}$.

Per-pixel prediction selection based on prediction confidence. Let $(k, n, c^*) = \arg\max_c \hat{y}_{k,n,c}$ and $(k, n, c') = \arg\max_{c \neq c^*} \hat{y}_{k,n,c}$. Then $\delta_{n,k} := (\hat{y}_{k,n,c^*} - \hat{y}_{k,n,c'})$, the difference between the largest and second-largest class probability is a measure of the predictive confidence of classifier k. We choose the most *confident* prediction ($k^* = \arg\max_k \delta_{n,k}$, so $c^* = \arg\max_c \hat{y}_{k^*,n,c}$) for pixel n as its final predicted label. We denote this strategy P_Dist.

Per-pixel prediction ensemble based on prediction confidence. To mitigate against a concern of an incorrect prediction made with high confidence, we propose P_Ens, a per-Pixel ensemble strategy that weights multiple predictions based on their confidence. Thus multiple less confident predictions can compensate in test cases where the highest confident prediction may be incorrect.

The detailed process of generating final prediction using P_Dist or P_Ens strategies is described in Appendix A.

4. Experiments and Results

4.1. Datasets

MoNuSeg dataset The MoNuSeg dataset (Kumar et al., 2019) is a multi-organ nucleus segmentation dataset. The training set includes 37 images of size 1000×1000 from 4 different organs (lung, prostate, kidney, and breast). The test set contains 14 images with more than 7000 nucleus boundary annotations. All images are scaned at 40× magnification. A 400×400 window slides through the images with a stride of 200 pixels to separate each image into 16 tiles for training and testing.

BCSS dataset The Breast Cancer Semantic Segmentation (BCSS) dataset (Amgad et al., 2019) consists of 151 H&E stained whole-slide images and ground truth masks corresponding to 151 histologically confirmed breast cancer cases. Tissue types of the BCSS dataset consists of 5 classes (i)tumour, (ii)stroma, (iii)inflammatory infiltration, (iv)necrosis and (v)others. We set aside slides from 7 institutions to create our test set and used the remaining images for training. Shift and crop data augmentation, random horizontal and vertical flip were adopted to enrich training samples. Finally, 3154 and 1222 pixel tiles of size 512×512 were cropped for training and testing, respectively.

4.2. Degree of equivariance

To quantitatively compare the degree to which our proposed method preserves scale equivariance relative to other scale-equivariant convolutional layers, we rescale N test images $f_i \mapsto S_s(f_i)$ by scale factor s, extract feature maps $\Phi(\cdot)$ and $\Phi'(\cdot)$, and then calculate the equivariance error:

$$\Delta_s = \frac{1}{N} \sum_{i=1}^{N} \frac{\|S_s \Phi(f_i) - \Phi'(S_s(f_i))\|_2^2}{\|S_s \Phi(f_i)\|_2^2}. \tag{10}$$

where $\Phi(\cdot)$ and $\Phi'(\cdot)$ denotes a sequence of convolutional operations with filters parameterised by different sets of σ_k.

4.3. Compared methods

We use the UNet architecture as a backbone and replace the conventional convolutional layers with different types of scale-equivariant convolution to generate 3 scale-equivariant UNet variants (SESN: the UNet with SESN layers; DISCO: the UNet with DISCO layers; SEUNet: the UNet with the proposed Gaussian derivative layers, the model generates γ segmentation maps). For the UNet with conventional convolutional layers, the number of filters at each depth are 60, 120, 240, 480, 960. For a fair comparison, all of UNet variants have the same number of scales (refers to the hyper-parameter γ). for the SESN and DISCO model, we set γ as 5 and scale factors as $\{1, 2, 3, 4, 5\}$, therefore the size of filters at each scale is $\{3, 5, 7, 9, 11\}$. For SESN model, we set the highest order of Hermite polynomial as 4 since it demonstrates the best performance. For the proposed models, we carefully set the lower and the upper bound of σ_k^l to set the size of filters (derived from equation (7)) of the first layer to be consistent with that of SESN and DISCO. The σ range of each layer is shown in Figure 3(a) and 3(b) (black dashed lines). We set the highest order of the Gaussian derivative to be 1, since using higher order derivatives fails to provide better performance. The colour normalisation method proposed in Vahadane et al. (2016) is used to remove stain colour variation, before training. All models are trained on images at the original scale, scale augmentation is not used in all of our experiments.

We implement the conventional UNet model and our proposed methods. The officially released source code of SESN and DISCO layers is used in our experiments. All Models are implemented in Pytorch Paszke et al. (2019) and trained on one NVIDIA RTX 8000 GPU using the Adam optimiser Kingma and Ba (2014) with weight decay of 10^{-4} to minimise the cross-entropy loss. The training epoch is set as 70, and the initial learning rate for

Test Scale		0.25	0.3	0.35	0.42	0.5	0.59	0.71	0.84	1
Pred Head	1	30.25	34.04	37.12	**43.99**	**52.49**	**57.71**	59.87	58.98	57.17
	2	**30.39**	**34.39**	**37.68**	43.75	49.78	56.32	**60.25**	**59.94**	**58.28**
	3	26.62	30.76	34.06	40.36	46.62	53.58	59.02	59.78	58.20
	4	22.56	26.27	29.21	34.47	40.58	49.14	56.56	58.97	57.82
	5	21.81	24.55	27.10	31.73	36.85	44.16	53.27	57.89	58.13
Test Scale		1.19	1.41	1.68	2	2.38	2.83	3.36	4	mean
Pred Head	1	51.90	43.62	34.72	27.41	23.66	21.23	19.89	19.13	39.60
	2	54.08	46.32	37.86	30.50	25.42	22.28	20.76	19.97	40.47
	3	**54.95**	47.82	39.62	32.96	27.99	23.64	21.76	20.93	39.92
	4	54.70	**48.94**	**41.68**	35.70	31.63	27.66	24.16	22.33	38.96
	5	54.62	49.11	41.31	**35.87**	**32.26**	**29.25**	**26.63**	**24.64**	38.19

Table 1: The mIoU score on the BCSS dataset (highest in bold). Head 1-5 denote 5 classifiers appended to the 5 groups of filters of increasing σ at the last layer. The actual σ values are shown in Figure 3(a). The last column is the mean mIoU score over all scales.

the Adam optimiser is set as 0.015 and then changed according to the 1cycle learning rate policy Smith and Topin (2019). The batch size is 20 for training models.

4.4. Results and Discussion

In this section, we report the overall segmentation performance of the three UNet architectures followed by ablation studies to analyse the performance gain of our approach. For the BCSS dataset, we use the mean Intersection over Union (mIoU) to measure segmentation performance of models, while for the binary task in the MoNuSeg dataset, we report the IoU score of the nuclei class. Examples of images, masks and segmentation maps generated by models can be seen in Appendix D.

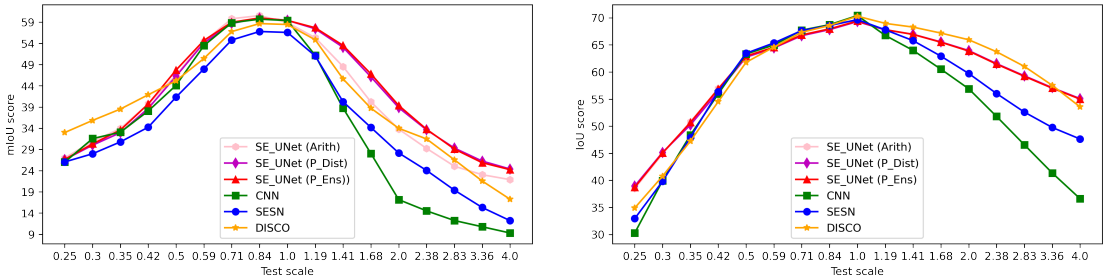
Evaluation regime. Since we aim to evaluate models' scale equivariance property, we re-scale the test set by a series of scale factors between 0.25 and 4, with a relative scale ratio of $\sqrt[4]{2}$ between adjacent testing scales.

Scale specific predictions for SEUNet. Table 1 summarises the mIoU score of the proposed method on re-scaled test images. Although we offer 3 strategies to generate the final segmentation maps from γ predictions, we first check the performance of each group of filters. As seen in the table, the best prediction head shifts to the one with larger σ values, as the scale of images increase (although prediction head 2 gives best prediction for scales between 0.25-0.35, the performance gap to the prediction head 1 is very small). This is consistent with our intuition that to capture the same information from enlarged images, filter sizes should also increase.

Comparison with CNN baseline and other equivariant methods. Figure 2(a) and 2(b) show the per-scale test performance of different approaches on the BCSS and the MoNuSeg datasets. The performance of all models are similar when the test scale is close to the original training scale. However, as the test scale moves away from the training scale,

Dataset	Metric	CNN	SESN	DISCO	Arithm	P_Dist	P_Ens
MoNuSeg	IoU	54.94	57.29	59.77	59.96	59.93	**59.98**
	E_Err	0.64	0.57	0.39		**0.26**	
BCSS	mIoU	34.48	35.36	40.51	40.51	42.10	**42.43**
	E_Err	0.74	0.72	0.61		**0.54**	

Table 2: Experimental results of different methods on MoNuSeg and BCSS dataset. E_Err denotes the equivariance error of feature maps generated by the last layer (before the final prediction generation step).



(a) BCSS dataset

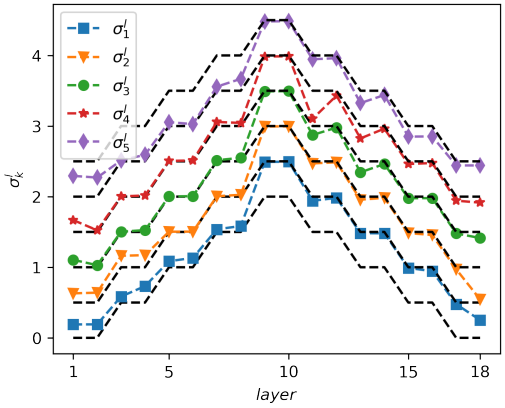
(b) MoNuSeg dataset

Figure 2: Comparison of the Per-scale mIoU score between models.

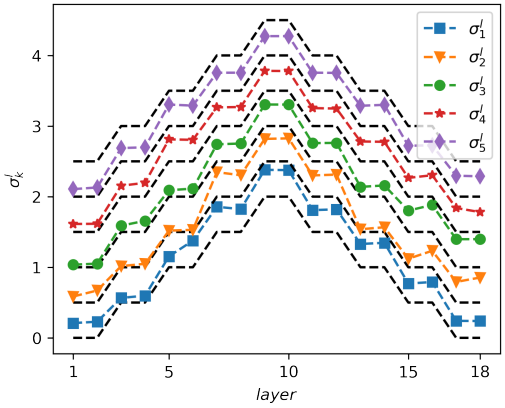
the performance of the conventional CNN drops significantly. Although the performance of SESN, DISCO and the proposed method also drop as the test scale changes, these models look more robust than the CNN. Table 2 compares models in terms of the averaged mIoU scores over different scales, the equivariance of feature map at the last layer, and the number of trainable parameters. As observed from the table, the proposed SEUNet with different prediction ensemble/selection strategies outperforms all compared methods, particularly on the BCSS dataset, while using fewer parameters. In terms of prediction strategy, simply averaging the prediction demonstrates the worst performance (on the BCSS dataset). This suggests that mixing features of all scales equally without considering the possibility that scale-specific filters have different contributions to the prediction is not the optimal choice. The proposed P_Dist strategy surpasses the arithmetic mean ensemble by 1.59 points on the BCSS dataset. Moreover, the P_Ens further boosts the performance by 0.33 points, when compared with the P_Dist. This comparison validates the effectiveness of the P_Dist and P_Ens strategy. We report the equivariance error Δ_s in Table 2 computed using the final layer outputs of all three networks on both datasets. We note that lower equivariance error correlates with better segmentation performance when tested on multi-scale images.

5. Ablation Study

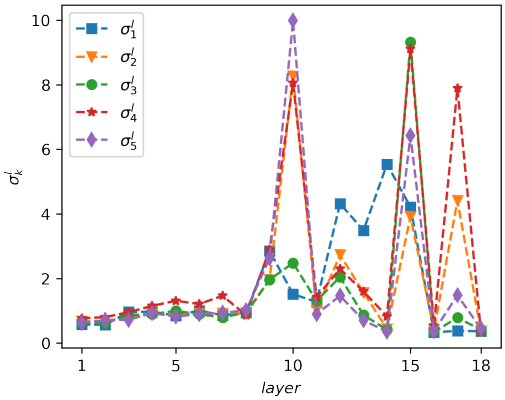
In section 3.2, we propose to constrain the value of σ_k^l in some non-overlapping ranges, to ensure that the constructed filters capture relevant patterns at different scales. Here, to ver-



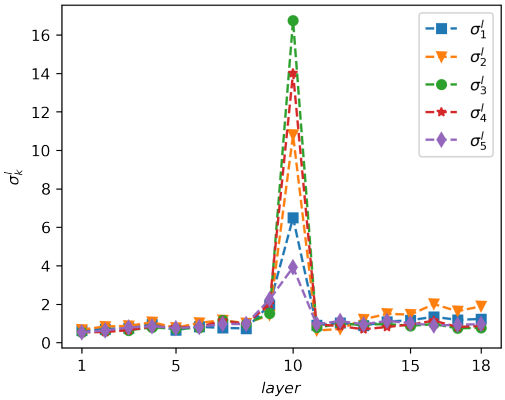
(*a*) σ_k^l is constrained (BCSS).



(*b*) σ_k^l is constrained (MoNuSeg).



(*c*) σ_k^l is trained freely (BCSS).



(*d*) σ_k^l is trained freely (MoNuSeg).

Figure 3: The σ_k^l values of filters in different layers. (a) and (b) The σ_k^l of models trained on the BCSS and the MoNuSeg datasets under range constraints. (c) and (d) The σ_k^l of models trained on the BCSS and the MoNuSeg datasets without imposing range constraints. The black dashed lines in (a) and (b) show the range of σ_k^l of layers.

ify the effectiveness of imposing range constraints and to trace the origin of the performance gain of the proposed method, we conduct the following two ablation experiments.

1) Fix σ_k^l values. Instead of constraining σ_k^l to the range of $(a_k^l,\ b_k^l)$ in equation (6), we fix the σ_k^l to be $\frac{a_k^l+b_k^l}{2}$.

2) σ_k^l being trained freely.

	MoNuSeg			BCSS		
	Cons	Fixed	Free	Cons	Fixed	Free
mIoU	**59.98**	58.36	57.40	**42.43**	41.45	39.02

Table 3: mIoU of SEUNets trained in different σ settings.

Table 3 summarises the performance achieved by models trained under different σ_k^l settings. The mIoU score reported in the table is the mean of per-scale mIoU score obtained by P_Ens. As shown in the table, models trained with range constraint outperform models with σ_k^l being fixed or trained completely freely, on both datasets. Figure 3 shows the σ_k^l values of models trained under different settings. As can be seen in Figure 3(c) and 3(d), allowing σ_k^l to be trained freely results in the case that multiple σ_k^ls converge to the same value. This is detrimental to the feature representation of the model since the same σ_k^l means that the same scale of the generated filters, thus the resultant feature map is also the same (because the coefficient α is shared between filters in different groups). Therefore, features are redundant and are not scale equivariant. This is the reason why the model trained without range constraint demonstrates the worst performance. The model trained with fixed σ_k^l values performs slightly better than the freely trained one, since σ_k^ls are non-overlapping, multi-scale filters can be constructed to extract information from different scaled images. However, the manually selected σ_k^ls may not be the optimal choice that fits the dataset best. Moreover, the optimal set of σ_k^ls may vary from dataset to dataset. This motivated our choice in equation (6) to train σ_k^ls to remain in disjoint intervals. As observed from Figure 3(a) and 3(b), σ_k^ls trained under constraint deviate from fixed values. And also, the learned σ_k^ls are quite different on the BCSS and the MoNuSeg datasets. Another benefit of imposing range constraints on σ_k^l is to reduce the computational complexity of the model. In our experiments, we observe that the model trained with range constraints required only $\frac{1}{3}$ the training time of the freely trained one. Because the filter of smaller size is less computationally intensive when performing convolution operations. For example, in Figure 3(a) and 3(b), the maximum σ_k^l values are 4.5 and 9.99 (layer 10), respectively, and the corresponding filter sizes are 19 and 41 (calculated from equation (7)). Therefore, the amount of computation required by the latter is $\approx 4.65\times$ that of the former when convolving with an image.

6. Conclusion

In this paper, we propose a Scale-Equivariant UNet (SEUNet) to address the challenge of generalising neural network segmentation on histopathology images to unseen scales. Firstly, we parameterise multi-scale filters by linearly combining groups of Gaussian derivative filters. The constructed filters are then used to learn scale-space representations that have a built-in scale-equivariant property. We constrain filter scales to be both trainable yet cover disjoint ranges. This is useful for finding dataset-adapted scale parameters. The extensive experimental results on two public datasets demonstrate that the proposed SEUNet achieves state-of-the-art performance. However, since we learn the Gaussian derivatives during training, these derivatives should be updated after each round of σ_k^l updating, which is more computationally expensive than using a pre-calculated filter basis. In the future, more range constraints will be explored to enable improved performance.

Acknowledgement

The authors acknowledge the use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this

work. The first author (Yilong Yang) is supported by China Scholarship Council under Grant No. 201906310150.

References

Mohamed Amgad, Habiba Elfandy, Hagar Hussein, Lamees A Atteya, Mai AT Elsebaie, Lamia S Abo Elnasr, Rokia A Sakr, Hazem SE Salem, Ahmed F Ismail, Anas M Saad, et al. Structured crowdsourcing enables convolutional segmentation of histology images. *Bioinformatics*, 35(18):3461–3467, 2019.

Ruqayya Awan, Korsuk Sirinukunwattana, David Epstein, Samuel Jefferyes, Uvais Qidwai, Zia Aftab, Imaad Mujeeb, David Snead, and Nasir Rajpoot. Glandular morphometrics for objective grading of colorectal adenocarcinoma histology images. *Scientific reports*, 7 (1):1–12, 2017.

Erik J Bekkers. B-spline cnns on lie groups. In *International Conference on Learning Representations*, 2019.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

Angjoo Kanazawa, Abhishek Sharma, and David Jacobs. Locally scale-invariant convolutional neural networks. *arXiv preprint arXiv:1412.5104*, 2014.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Iasonas Kokkinos. Pushing the boundaries of boundary detection using deep learning. *arXiv preprint arXiv:1511.07386*, 2015.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Neeraj Kumar, Ruchika Verma, Deepak Anand, Yanning Zhou, Omer Fahri Onder, Efstratios Tsougenis, Hao Chen, Pheng-Ann Heng, Jiahui Li, Zhiqiang Hu, et al. A multi-organ nucleus segmentation challenge. *IEEE transactions on medical imaging*, 39(5):1380–1391, 2019.

Jiahui Li, Shuang Yang, Xiaodi Huang, Qian Da, Xiaoqun Yang, Zhiqiang Hu, Qi Duan, Chaofu Wang, and Hongsheng Li. Signet ring cell detection with a semi-supervised learning framework. In *International conference on information processing in medical imaging*, pages 842–854. Springer, 2019.

Tony Lindeberg. Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics*, 21(1-2):225–270, 1994.

Tony Lindeberg. Scale-covariant and scale-invariant gaussian derivative networks. *Journal of Mathematical Imaging and Vision*, 64(3):223–242, 2022.

Diego Marcos, Benjamin Kellenberger, Sylvain Lobry, and Devis Tuia. Scale equivariance in cnns with vector fields. *arXiv preprint arXiv:1807.11783*, 2018.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

Silvia L Pintea, Nergis Tömen, Stanley F Goes, Marco Loog, and Jan C van Gemert. Resolution learning in deep convolutional networks using scale-space theory. *IEEE Transactions on Image Processing*, 30:8342–8353, 2021.

Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, page 1100612. International Society for Optics and Photonics, 2019.

Ivan Sosnovik, Michał Szmaja, and Arnold Smeulders. Scale-equivariant steerable networks. In *International Conference on Learning Representations*, 2020.

Ivan Sosnovik, Artem Moskalev, and Arnold Smeulders. Disco: accurate discrete scale convolutions. In *British Machine Vision Conference*, 2021a.

Ivan Sosnovik, Artem Moskalev, and Arnold Smeulders. How to transform kernels for scale-convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 1092–1097, October 2021b.

Abhishek Vahadane, Tingying Peng, Amit Sethi, Shadi Albarqouni, Lichao Wang, Maximilian Baust, Katja Steiger, Anna Melissa Schlitter, Irene Esposito, and Nassir Navab. Structure-preserving color normalization and sparse stain separation for histological images. *IEEE transactions on medical imaging*, 35(8):1962–1971, 2016.

Daniel Worrall and Max Welling. Deep scale-spaces: Equivariance over scale. *Advances in Neural Information Processing Systems*, 32, 2019.

Yichong Xu, Tianjun Xiao, Jiaxing Zhang, Kuiyuan Yang, and Zheng Zhang. Scale-invariant convolutional neural networks. *arXiv preprint arXiv:1411.6369*, 2014.

Wei Zhu, Qiang Qiu, Robert Calderbank, Guillermo Sapiro, and Xiuyuan Cheng. Scaling-translation-equivariant networks with decomposed convolutional filters. *Journal of Machine Learning Research*, 23(68):1–45, 2022.

Appendix A. Pseudo-Code of Generating Final Prediction

We first calculate the prediction confidence, the difference between the largest and the second largest probability, then the weighting factor w_k of prediction $\hat{y}_k$ is determined by the *softmax* function, giving larger weights to confident predictions.

Algorithm 1: Per-pixel Prediction Selection/Ensemble based on Probability Distance.

Input: Probability vectors $[\hat{y}_1, \cdots, \hat{y}_\gamma]$, $\hat{y}_k \in \mathbb{R}^C$, $k \in \{1, \cdots, \gamma\}$; C is the number of classes. $Strategy \in \{P_Dist$ or $P_Ens\}$.

$D = [\,]$; `// Recording distance.`
for $k = 1, 2, ..., \gamma$ **do**
  $p_{max} = \max(\hat{y}_k)$
  $p_{max_idx} = \arg\max(\hat{y}_k)$
  $\hat{y}_{k,p_{max_idx}} = -1$
  $p_{second_max} = \max(\hat{y}_k)$
  $D.\text{append}(p_{max} - p_{second_max})$
  $\hat{y}_{k,p_{max_idx}} = p_{max}$
end
if *Strategy==P_Dist* **then**
  $k = \arg\max D$
  $\hat{y} = \arg\max_C(\hat{y}_k)$
end
if *Strategy==P_Ens* **then**
  for $k = 1, 2, ..., \gamma$ **do**
    $w_k = \frac{e^{D_k}}{\sum_{k=1}^{\gamma} e^{D_k}}$
  end
  $\hat{y} = \arg\max_C(w_k \hat{y}_k)$
end

Output: $\hat{y}$

Appendix B. Visualisation of Multi-Scale Filters

Given a set of scale parameters $\{\sigma_1, \cdots, \sigma_\gamma\}$ and the number of scales γ, the filter of each scale can be constructed by:

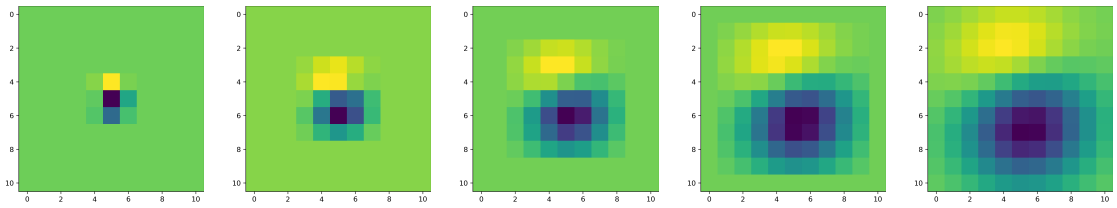
$$F_k(x, y; \sigma_k) = \sum_{0 \le i,\, 0 \le j}^{i+j \le N} \alpha_{i,j} \frac{\partial^{i+j}}{\partial x^i \partial y^j} G(x, y; \sigma_k),\ k \in \{1, \cdots, \gamma\} \tag{11}$$

Here we visualise the multi-scale filters generated with a set of predefined σ and randomly initialised coefficients (α). As shown in Figure 4, filters are similar in shape but vary in scale.

Appendix C. Visualising the Equivariance Error

To demonstrate the effectiveness of lowering equivariance error by convolving images with multi-scale filters, we convolve images at different scales with γ filters, paring feature maps and then calculate the equivariance error as:

$$\Delta_{s,k,k\prime} = \frac{1}{N} \sum_{i=1}^{N} \frac{\|S_s(F_k \star f_i) - F_{k'} \star S_s(f_i)\|_2^2}{\|S_s(F_k \star f_i)\|_2^2},\ s \in \mathbb{R}^+,\ k, k' \in \{1, \cdots, \gamma\} \tag{12}$$



$(a)\ \sigma_1 = 0.5$ $(b)\ \sigma_2 = 1$ $(c)\ \sigma_3 = 1.5$ $(d)\ \sigma_4 = 2$ $(e)\ \sigma_5 = 2.5$

Figure 4: Constructed multi-scale filters. From left to right, the effective size of the filter gradually enlarges as the σ increases.

where f_i is an image, F_k and $F_{k'}$ are filters with scale parameters σ_k and $\sigma_{k'}$, S_s is a scaling operation with factor s. Thus, given γ filters and two images at different scales, we arrive at a $\gamma \times \gamma$ equivariance error matrix. Where each element represents the equivariance error between feature maps, which are obtained by convolving images of different scales with different filters. As shown in Figure 5, for the feature map pair that produces the maximal matching, the ratio of scales between images is equal (or close) to the ratio of σ_ks between filters. For example, in Figure 5(p), the ratio between σs and the ratio between image scales is the same ($\frac{2}{0.5} = \frac{4}{1}$). The same phenomenon can be observed from images re-scaled by factors of 0.5 and 2 (Figure 5(e) and 5(l)). For images whose scales are not divisible, the matching degree between feature maps obtained by convolving the filter with the σ ratio closest to the image ratio is the highest. For example, in Figure 5(f), the ratio between images ($\frac{1}{0.59} \approx 1.69$) is close to the ratio between σs ($\frac{2.5}{1.5} \approx 1.67$). Thus, we experimentally validated that the scale equivarance err can be reduced by convolving images at different scales with appropriate filters whose scale is corresponded to the scale of images.

Appendix D. Visualisation of Model Prediction

To better understand the SEUNet, we visualise segmentation maps generated by the SEUNet and other compared models on input images at different scales. As shown in Figure 6 and 7, the SEUNet can retain a relative decent prediction when compared with other methods.

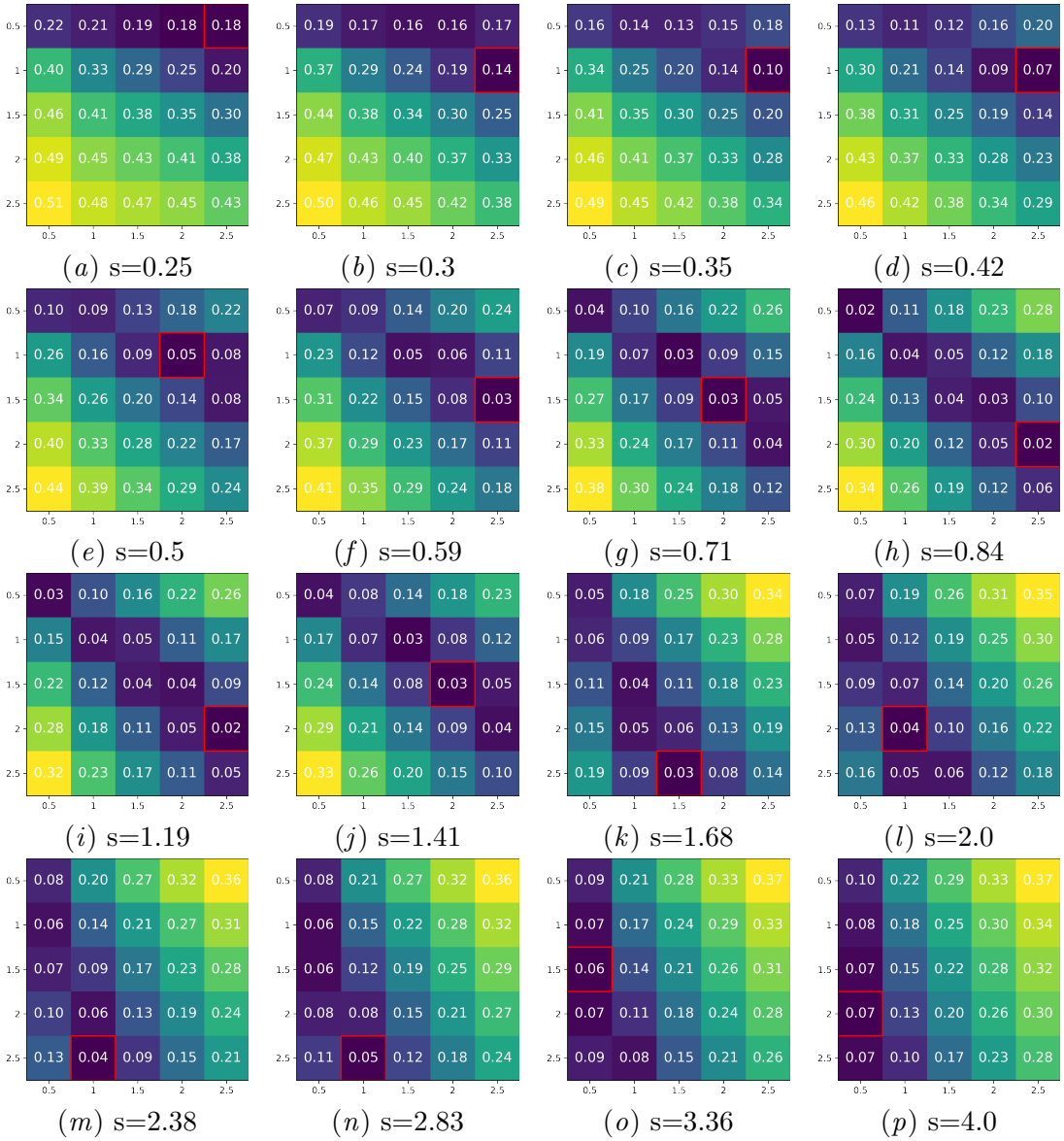


Figure 5: Scale equivariance error of feature maps. Each plot shows the equivariance errors between feature maps of the original image and the re-scaled image. In the title of each plot, s denotes the scale factor. The x-axis and the y-axis of each plot denote the σ of the filter that is used to convolve with the original image and the re-scaled image, respectively. In each plot, the number on the grid denotes the equivariance error between feature maps $S_s(F_k \star f)$ and $F_{k'} \star S_s(f)$. The lowest equivariance error in each 5×5 error matrix is highlighted by a red box.

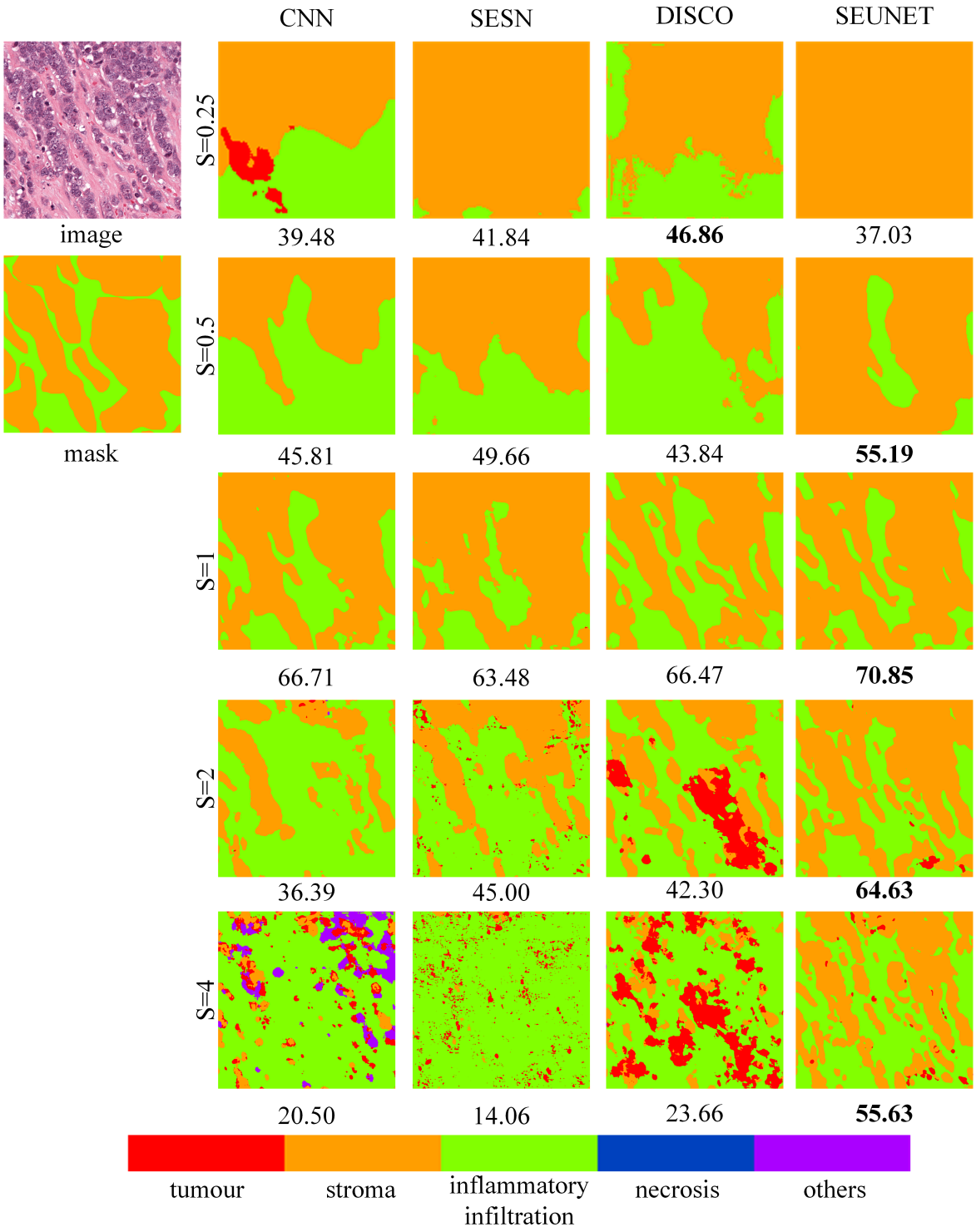


Figure 6: Visual comparison on the BCSS dataset. The mIoU score of each prediction is reported below the segmentation map. The highest score is highlighted in bold. Each column shows segmentation maps of a model on an image re-scaled by different scaling factors ($S = 1$ denotes the original scale).

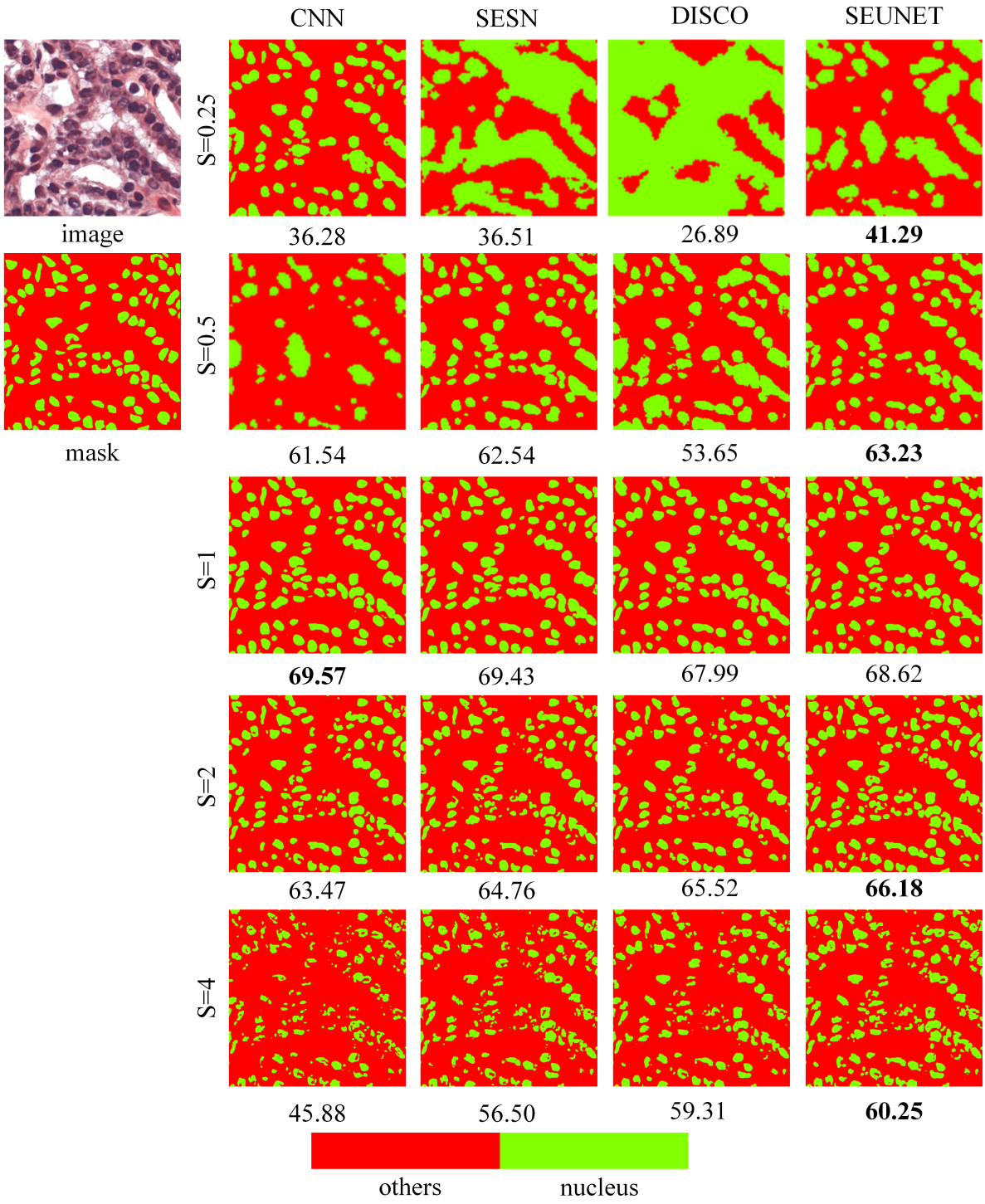


Figure 7: Visual comparison on the MoNuSeg dataset. The IoU score of each prediction is reported below the segmentation map. The highest score is highlighted in bold. Each column shows segmentation maps of a model on an image re-scaled by different scaling factors ($S = 1$ denotes the original scale).